Variable lookup

$$\langle \frac{x}{v} \cdots\rangle_{\mathsf{k}}\ \langle \cdots x \mapsto l \cdots\rangle_{\mathsf{env}}\ \langle \cdots l \mapsto v \cdots\rangle_{\mathsf{store}}$$

Blocks

$$\langle \frac{\{\texttt{vars}\, xl\,;\, s\}}{s \curvearrowright \mathsf{env}(\rho)} \cdots\rangle_{\mathsf{k}}\ \langle \frac{\rho}{\rho[n\,..\,n +_{Int} |xl| -_{Int} 1/xl]}\rangle_{\mathsf{env}}\ \langle\frac{\sigma}{\sigma[0/n\,..\,n+_{Int}|xl| -_{Int} 1]}\rangle_{\mathsf{store}}\ \langle\frac{n}{n +_{Int} |xl|}\rangle_{\mathsf{nextLoc}}$$

$$\langle \frac{\mathsf{env}(\rho)}{\cdot} \cdots\rangle_{\mathsf{k}}\ \langle \frac{\_}{\rho}\rangle_{\mathsf{env}}$$

Side effects: increment

$$\langle \frac{\texttt{++}\,x}{i +_{Int} 1} \cdots\rangle_{\mathsf{k}}\ \langle \cdots x \mapsto l \cdots\rangle_{\mathsf{env}}\ \langle \cdots l \mapsto \frac{i}{i +_{Int} 1} \cdots\rangle_{\mathsf{store}}$$

References

$$\langle \frac{\texttt{\&}\,x}{l} \cdots\rangle_{\mathsf{k}}\ \langle \cdots x \mapsto l \cdots\rangle_{\mathsf{env}}$$

$$\langle \frac{\texttt{*}\,l}{v} \cdots\rangle_{\mathsf{k}}\ \langle \cdots l \mapsto v \cdots\rangle_{\mathsf{store}}$$

$$\langle \frac{\texttt{malloc}\,(n)}{l} \cdots\rangle_{\mathsf{k}}\ \langle \cdots \frac{\cdot}{l \mapsto n} \cdots\rangle_{\mathsf{ptr}}\ \langle \frac{\sigma}{\sigma[0/l\,..\,l +_{Int} n -_{Int} 1]}\rangle_{\mathsf{store}}\ \langle \frac{l}{l +_{Int} n}\rangle_{\mathsf{nextLoc}}$$

$$\langle \frac{\texttt{free}\,(l)}{\cdot} \cdots\rangle_{\mathsf{k}}\ \langle \cdots \frac{l \mapsto n}{\cdot} \cdots\rangle_{\mathsf{ptr}}\ \langle \frac{\sigma}{\sigma[\bot/l\,..\,l +_{Int} n -_{Int} 1]}\rangle_{\mathsf{store}}$$

Exercise 180. *Evaluating an expression and returning a reference to it:*

$$\texttt{ref} \equiv \lambda x.(\lambda p.\texttt{*}\,p \,\texttt{:=}\, x \,;\, p)(\texttt{malloc}\,(1))$$

$$\langle \frac{\texttt{ref}\; v}{l} \cdots\rangle_{\mathsf{k}}\ \langle \frac{\sigma}{\sigma[v/l]}\rangle_{\mathsf{store}}\ \langle \frac{l}{l +_{Int} 1}\rangle_{\mathsf{nextLoc}}$$

Assignment

$$\langle \frac{x \,\texttt{:=}\, v}{\cdot} \cdots\rangle_{\mathsf{k}}\ \langle \cdots x \mapsto l \cdots\rangle_{\mathsf{env}}\ \langle \cdots l \mapsto \frac{\_}{v} \cdots\rangle_{\mathsf{store}}$$

context: $\texttt{*}\,e \,\texttt{:=}\, e'$ $[\mathit{strict}(e)]$

$$\langle \frac{\texttt{*}\,l \,\texttt{:=}\, v}{\cdot} \cdots\rangle_{\mathsf{k}}\ \langle \cdots l \mapsto \frac{\_}{v} \cdots\rangle_{\mathsf{store}}$$

Function and aspects

$$\langle \frac{\texttt{aspect}\, s}{\cdot} \cdots \rangle_{\mathsf{k}}\ \langle \frac{\_}{s} \rangle_{\mathsf{aspect}}$$

$$\langle \frac{\lambda xl\, .\, e}{closure(xl, s \curvearrowright e, \rho)} \cdots \rangle_{\mathsf{k}}\ \langle \rho \rangle_{\mathsf{env}}\ \langle s \rangle_{\mathsf{aspect}}$$

Exercise 181. *Define a different semantics for aspects, or a new aspect construct, that would weave the aspect at function call time instead of at function declaration time.*

Function application

$$\langle \frac{closure(xl, e, \rho)\ vl \cdots}{e \curvearrowright \mathsf{env}(\varrho)} \rangle_{\mathsf{k}}\ \langle \frac{\varrho}{\rho[n \mathrel{..} n +_{Int} |xl| -_{Int} 1 / xl]} \rangle_{\mathsf{env}}\ \langle \frac{\sigma}{\sigma[vl / n \mathrel{..} n +_{Int} |xl| -_{Int} 1]} \rangle_{\mathsf{store}}\ \langle \frac{n}{n +_{Int} |xl|} \rangle_{\mathsf{nextLoc}}$$

$$\langle vl \curvearrowright \frac{\mathsf{env}(\rho)}{\cdot} \cdots \rangle_{\mathsf{k}}\ \langle \frac{\_}{\rho} \rangle_{\mathsf{env}}$$

Tail call optimization:

$$\frac{\mathsf{env}(\_)}{\cdot} \curvearrowright \mathsf{env}(\_)$$

Exercise 182. *Prove that the tail call optimization is sound.*

Recursion

$$\langle \frac{\mu x\, .\, e}{e \curvearrowright \mathsf{env}(\rho)} \cdots \rangle_{\mathsf{k}}\ \langle \frac{\rho}{\rho[n/x]} \rangle_{\mathsf{env}}\ \langle \cdots \frac{\cdot}{n \mapsto \mu x\, .\, e} \cdots \rangle_{\mathsf{store}}\ \langle \frac{n}{n +_{Int} 1} \rangle_{\mathsf{nextLoc}}$$

Call with current continuation (call/cc)

$$\langle \frac{\texttt{callcc}\ v}{v\ cc(k, \rho)} \curvearrowright k \rangle_{\mathsf{k}}\ \langle \rho \rangle_{\mathsf{env}}$$

$$\langle \frac{cc(k, \rho)\ v \cdots}{v \curvearrowright k} \rangle_{\mathsf{k}}\ \langle \frac{\_}{\rho} \rangle_{\mathsf{env}}$$

Sequential non-determinism

$$\langle \frac{\texttt{randomBool}}{true} \cdots \rangle_{\mathsf{k}}$$

$$\langle \frac{\texttt{randomBool}}{false} \cdots \rangle_{\mathsf{k}}$$

Threads

$$\langle \cdots \langle \frac{\texttt{spawn}\ s}{\cdot} \cdots \rangle_{\mathsf{k}}\ \langle \rho \rangle_{\mathsf{env}} \cdots \rangle_{\mathsf{thread}}\ \frac{\cdot}{\langle \langle s \rangle_{\mathsf{k}}\ \langle \rho \rangle_{\mathsf{env}}\ \langle \cdot \rangle_{\mathsf{holds}} \rangle_{\mathsf{thread}}}$$

$$\frac{\langle \cdots \langle \cdot \rangle_{\mathsf{k}}\ \langle h \rangle_{\mathsf{holds}} \cdots \rangle_{\mathsf{thread}}}{\cdot}\ \langle \frac{busy}{busy -_{set} dom(h)} \rangle_{\mathsf{busy}}$$

Thread synchronization

$$\langle \frac{\texttt{acquire}\ v}{\cdot} \cdots \rangle_{\mathsf{k}}\ \langle \cdots v \mapsto \frac{n}{s_{Nat}(n)} \cdots \rangle_{\mathsf{holds}}$$

$$\langle \frac{\texttt{acquire}\ v}{\cdot} \cdots \rangle_{\mathsf{k}}\ \langle \cdots \frac{\cdot}{v \mapsto 0} \cdots \rangle_{\mathsf{holds}}\ \langle busy\ \frac{\cdot}{v} \rangle_{\mathsf{busy}} \quad \text{when } \neg_{Bool}(v \in_{set} busy)$$

$$\langle \frac{\texttt{release}\ v}{\cdot} \cdots \rangle_{\mathsf{k}}\ \langle \cdots v \mapsto \frac{s_{Nat}(n)}{n} \cdots \rangle_{\mathsf{holds}}$$

$$\langle \frac{\texttt{release}\ v}{\cdot} \cdots \rangle_{\mathsf{k}}\ \langle \cdots \frac{v \mapsto 0}{\cdot} \cdots \rangle_{\mathsf{holds}}\ \langle \cdots \frac{v}{\cdot} \cdots \rangle_{\mathsf{busy}}$$

Rendez-vous synchronization

$$\langle \frac{\texttt{rv}\ v}{\cdot} \cdots \rangle_{\mathsf{k}}\ \langle \frac{\texttt{rv}\ v}{\cdot} \cdots \rangle_{\mathsf{k}}$$

Agents

$$\langle \frac{\texttt{new-agent}\ s}{n} \cdots \rangle_{\mathsf{k}}\ \langle m \rangle_{\mathsf{me}}\ \langle \frac{n}{n +_{Int} 1} \rangle_{\mathsf{nextAgent}}\ \frac{\cdot}{NewAgent}$$

where $NewAgent$ stands for:

$$\langle \langle \langle s \rangle_{\mathsf{k}}\ \langle \cdot \rangle_{\mathsf{env}}\ \langle \cdot \rangle_{\mathsf{holds}} \rangle_{\mathsf{thread}}\ \langle \cdot \rangle_{\mathsf{busy}}\ \langle n \rangle_{\mathsf{me}}\ \langle m \rangle_{\mathsf{parent}}\ \langle \cdot \rangle_{\mathsf{store}}\ \langle 0 \rangle_{\mathsf{nextLoc}}\ \langle \cdot \rangle_{\mathsf{aspect}} \rangle_{\mathsf{agent}}$$

$$\langle \cdots \langle \cdot \rangle_{\mathsf{threads}} \cdots \rangle_{\mathsf{agent}} \rightharpoonup \cdot$$

$$\langle \frac{\texttt{me}}{m} \cdots \rangle_{\mathsf{k}}\ \langle m \rangle_{\mathsf{me}}$$

$$\langle \frac{\texttt{parent}}{n} \cdots \rangle_{\mathsf{k}}\ \langle n \rangle_{\mathsf{parent}}$$

$$\langle \frac{\texttt{send-asynch}\ n\ v}{\cdot} \cdots \rangle_{\mathsf{k}}\ \langle m \rangle_{\mathsf{me}}\ \frac{\cdot}{\langle \langle m \rangle_{\mathsf{sender}}\ \langle n \rangle_{\mathsf{receiver}}\ \langle v \rangle_{\mathsf{val}} \rangle_{\mathsf{message}}}$$

$$\langle \frac{\texttt{receive-from}\ n}{v} \cdots \rangle_{\mathsf{k}}\ \langle m \rangle_{\mathsf{me}}\ \frac{\langle \langle n \rangle_{\mathsf{sender}}\ \langle m \rangle_{\mathsf{receiver}}\ \langle v \rangle_{\mathsf{val}} \rangle_{\mathsf{message}}}{\cdot}$$

$$\langle \underset{v}{\underline{\texttt{receive}}} \cdots\rangle_{\mathsf{k}} \langle m\rangle_{\mathsf{me}} \underset{\cdot}{\underline{\langle \cdots \langle m\rangle_{\mathsf{receiver}} \langle v\rangle_{\mathsf{val}} \cdots\rangle_{\mathsf{message}}}}$$

$$\langle \cdots \langle \underset{\cdot}{\underline{\texttt{send-synch}\ n\ v}} \cdots\rangle_{\mathsf{k}} \langle m\rangle_{\mathsf{me}} \cdots\rangle_{\mathsf{agent}} \langle \cdots \langle \underset{v}{\underline{\texttt{receive-from}\ m}} \langle n\rangle_{\mathsf{me}} \cdots\rangle_{\mathsf{k}} \cdots\rangle_{\mathsf{agent}}$$

$$\langle \underset{\cdot}{\underline{\texttt{send-synch}\ n\ v}} \cdots\rangle_{\mathsf{k}} \langle \cdots \langle \underset{v}{\underline{\texttt{receive}}} \cdots\rangle_{\mathsf{k}} \langle n\rangle_{\mathsf{me}} \cdots\rangle_{\mathsf{agent}}$$

Exercise 183. *While the rules for asynchronous sending allow the sender and the receiver to be same agent, the two rules for synchronous sending assume that the sender and the receiver agents are different. Add two other rules for synchronous sending that extend the semantics above to allow agents to communicate synchronously with themselves (they would have to send from one thread and receive from another thread).*

Abrupt termination

$$\langle \texttt{halt-thread} \cdots\rangle_{\mathsf{k}} \rightarrow \langle \cdot\rangle_{\mathsf{k}}$$

$$\langle \cdots \langle \texttt{halt-agent} \cdots\rangle_{\mathsf{k}} \cdots\rangle_{\mathsf{threads}} \rightarrow \langle \cdot\rangle_{\mathsf{threads}}$$

$$\langle \cdots \langle \texttt{halt-system} \cdots\rangle_{\mathsf{k}} \cdots\rangle_{\mathsf{agents}} \rightarrow \langle \cdot\rangle_{\mathsf{agents}}$$